|  |
| --- |
| **TEST STRATEGY DOCUMENT** |
|  |
| FARM-RISE APPLICATION |
| **AUTHOR NAME: Harish Takle** |
| **Date Written: 21/11/2018** |

| **Revision Number** | **Revision Date** | **Author(s)** | **Revision Description** |
| --- | --- | --- | --- |
| Version 1.0 | 20-11-2017 | QA TEAM | Newly Implemented features with MDM Services |
| Version 2.0 | 28-12-2017 | QA TEAM | Newly Implemented Features. |

**Farm-Rise Company**

Contents

1. INTRODUCTION 3

1.1. Description of Testing 3

1.2. Definitions, Acronyms, and Abbreviations 3

1.3 References / Project Artifacts 3

2. RESOURCE REQUIREMENTS 3

2.1 Testing Environment 3

2.2. Testing Tools 3

2.3. Project Staffing 3

3. TESTING SCOPE 4

3.1. Levels of Testing 4

3.1.1 Functional 4

3.1.2 Regression 4

3.1.3 Production 4

3.1.4 Performance 4

3.2 Areas Not Being Tested 4

3.3 Risks 5

4. STANDARDS AND METHODS 5

4.1 Reviews 5

4.2 Build Control 5

4.1. Defect Reporting 5

4.2. Procedure Controls 5

5. OPEN ISSUES 6

6. APPROVALS 7

7. APPENDIX A - Definitions, Acronyms, and Abbreviations 8

8. Document Tracking 9

# INTRODUCTION

## Description of Testing

This section should be used to define what is being tested and the primary purpose (the “why” the testing is being conducted. Consideration may be given to special circumstances, special focus/emphasis, or other issues that are unique to this project.

## Definitions, Acronyms, and Abbreviations

See Appendix A.

## 1.3 References / Project Artifacts

This section provides a list of all documents associated with this project. An example chart is provided below:

| ***Reference*** | ***File Name or URL*** |
| --- | --- |
| *Requirements* | *PDD Document PDD.doc* |
| *Defect Tracking (Bugzilla)* | [*http://tools.farmrise.com/bugzilla/buglist.cgi*](http://tools.farmrise.com/bugzilla/buglist.cgi) |
| *Test Strategy* | *R-1.0 TestStrategy.doc* |
| *Test Data Scenarios* | *R-1.0 Testing.xlsx* |
| *QA Results Memo* | *QM Report QM.doc* |

# 

# 2. RESOURCE REQUIREMENTS

## 2.1 Testing Environment

This section will describe the hardware and software necessary for the test environment in order to begin testing for this project.

## 2.2. Testing Tools

This section will describe the tools necessary to conduct the test (excluding manual tests).

## 2.3. Project Staffing

This section is used to identify key individuals involved with the test and their designated responsibility and availability. The table may be expanded to accommodate additional responsible parties.

|  |  |  |
| --- | --- | --- |
| **Area of Responsibility** | **Name(s)** | **Availability / Scheduling Constraints\*** |
| Pricing of a Corp | Test Member 1 | 2 weeks |
| Setting Module | Test Member 2 | 3 weeks |
| Corps Advisory Information | Test Member3 | 2 weeks |
| Weather and News Report |  |  |

# 

# 3. TESTING SCOPE

## 3.1. Levels of Testing

This section lists out the levels of testing that will be performed for this project.

* **Unit Testing** 
  + xxxxx
* **QA Testing**
  + Functional
  + Regression
* **User Acceptance Production Testing**
  + Alpha Testing (Before Release to Market)
  + Beta Testing (After releasing to Market)

This section describes the areas that will be covered during QA Testing.

### 3.1.1 Functional

This section describes the functional requirements that will be tested

### 3.1.2 Regression

This section describes the regression testing that will be conducted.

### 3.1.3 Production

This section describes the testing that will be conducted to uncover any production issues.

### 3.1.4 Performance

Refer to the separate Performance Test Strategy document located at: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

## 3.2Areas Not Being Tested

This section describes specific areas that will not be tested. A sample chart is provided below.

| **Area** | **Description Of What Will Not Be Tested** |
| --- | --- |
| XXXXX | * xxxxx * xxxxx |

## 3.3 Risks

This section outlines the risks and contingency plans associated with the testing phase. A sample chart is included below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***#*** | ***Risks*** | ***Prob. of Occurrence*** | ***Severity of Impact*** | ***Contingency Plan*** |
| *1* | *Lack of finalized specifications* | *High* | *High* | *Quality Assurance will perform Exploratory Testing.* |

# 4. STANDARDS AND METHODS

## 

## 4.1 Reviews

This section describes the reviews that will be conducted, who will conduct them, and when they will be conducted.

## 4.1.1. Build Control

This section describes who will be responsible for the build(s), what approval is required, and the process by which it will be completed.

## 4.1.2 Defect Reporting

This section outlines how defects and issues will be tracked. And how team members will manage the log and resolve the defect.

## 4.2. Procedure Controls

This section describes the procedure controls (initiation, critical failure, resumption, and completion) for this type of testing. The following chart serves as an example and should be updated to reflect this project.

| **Control** | **Description** |
| --- | --- |
| Initiation | Guidelines that must be met in order to start testing. The initiation controls are:   * Requirements/Scope document created and signed off by management. * Unit testing has been completed. * Product available and running in test environment. * Test Strategy created and signed off by management. |
| Critical Failure | Guidelines that determine the point at which a failure is deemed to be critical and testing will stop. A defect is not necessarily a critical failure. A critical failure is a defect or issue so severe that there is no point in continuing. Example: The Critical Failure controls are:   * System cannot be installed (critical). * System cannot be accessed (critical). |
| Resumption | Guidelines that determine the point at which testing can resume after resolution of a critical failure. Resumption controls are:   * Failure resolved and new release moved to test environment by development. |
| Completion | Guidelines that must be met for testing to be considered complete. Completion controls are:   * All high priority defects/issues have been resolved. * All defects/issues with been reported and addressed in some manner. * Once all testing has been completed, QA will issue a QA Results Memo to all involved parties. The memo will briefly describe the overall testing that was done, any open defects/issues with their severity and the final status of the testing (accepted, conditionally accepted, or not accepted for production). |

# 5. OPEN ISSUES

This section provides the location of the team’s issue log and instruction on how issues are managed and resolved.

# 6. APPROVALS

This section defines the individuals who have approval authority during the performance testing process for this project.

| **Name** | **Title** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 

# 7. APPENDIX A - Definitions, Acronyms, and Abbreviations

The chart below defines the various terms that will be used in this document as in communication related to the performance test. This list can be modified in any way to ensure that it reflects the terms of the specific project.

| **Term (Acronym)** | **Definition** |
| --- | --- |
| Test Case (TC) | A documented description of the inputs, execution instructions, and expected results, which are created for the purpose of determining whether a specific software feature works correctly or a specific requirement, has been satisfied. |
| Defect | For purposes of testing, a defect is defined as an anomaly caused by the system not functioning exactly as outlined in the requirements or the intended system functionality cannot be explicitly understood from the requirements and design documentation. |
| Revision Control | Sequential capturing of changes to an artifact that allows retracing (if necessary). Usually accomplished through the use of a tool. |
| Unit Testing | Unit testing is performed against a specific program by the developer who wrote it to test their own code and ensure that the program will operate according to the design specification. Usually executed independently of other programs, in a standalone manner. |
| Integration / System Testing | Integration testing is performed to demonstrate that the unit-tested programs work properly with each other when they are progressively assembled together to eventually operate as a cohesive, integrated system.  System testing is performed against the complete application system to demonstrate that it satisfies the User and Technical Requirements, within the constraints of the available technology. System testing is usually performed in conjunction with Integration Testing. |
| Functional Testing | Functional testing is performed in a specific testing environment, similar to production that verifies the functionality of the entire system, as it would be in a live environment. Testing efforts and objectives will center around test cases specifically derived from the requirements. In addition to specified error processing. Tests will be documented using formal test cases. |
| Regression Testing | Regression testing is performed to verify that the new code did not break any of the existing code. |
| Performance Testing | Performance testing is performed to verify how well the application measures up under varying loads of data, but still within the limits of normal, acceptable operating conditions. |
| Load Testing | Load testing is performed to demonstrate how the product functions under certain high volume conditions (helps determine its breaking point). Load testing is usually performed in conjunction with Performance Testing. |
| User Acceptance Testing | User Acceptance testing is performed to help validate the functionality of the entire system, including the manual procedures, which is usually performed by the system end-users. This testing helps insure that the system meets all the business scenarios that were identified by the users. |
| Automated Testing | Automated testing is performed to help validate the functionality of the entire system in a more efficient manner than manual testing. Regression testing will utilize the automated testing efforts. Currently, the tool used in automated testing is Segue Silk Test. |
| Quick Test Professional | Functional Automated testing tool. |
| LoadRunner | Performance/Load testing tool. |

# 8. Document Tracking

|  |  |  |
| --- | --- | --- |
| **Date** | **Action Taken** | **By Whom** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |